Dynamic Eigenvalue/Eigenvector Tracking Using
Continuous Optimization on Constraint Manifolds

Edwin B. Dean
NASA Langley Research Center
Mail Stop 430
Hampton, VA 23665
(804) 865-4894

Presented at the
The TIMS/ORSA Joint National Meeting
Denver Colorado

October 24-26, 1988

Abstract

This paper demonstrates the capability to obtain and dynamically track eigenvalues and eigenvectors using continuous optimization on constraint manifolds.

Introduction

The author has demonstrated [1] the capability of continuous optimization on constraint manifolds (COCM) [2-7] to obtain both numerical and closed form analytical solutions to optimization problems with C^k constraints and objective function. This paper displays another useful capability of COCM, that of deriving algorithm structures. The ability to formulate the eigenvalue/eigenvector problem as an optimization problem leads to algorithm structures for obtaining the eigenvalues and eigenvectors of a matrix A

To be consistent with the illuminating tensor notation of Gerretsen [8], row vector notation is used throughout this paper. Thus, a vector v with components a_i in basis $\{b_1, \ldots, b_n\}$ may be expressed as

$$v = a\,B$$

where

$$a = [a_1 \ldots a_n] \qquad \text{and} \qquad B = \begin{bmatrix} b_1 \\ \ldots \\ b_n \end{bmatrix}$$

All computer solutions were performed on a Macintosh Plus ™ using Lightspeed Pascal ™.

Summary of COCM

The following theory is based on concepts from differentiable manifolds and differential geometry as described by Thorpe [9] and Boothby [10].

An n-dimensional manifold is a connected, locally compact space with a countable basis, each point of which has a neighborhood homeomorphic to euclidian n-space.

A C^k differentiable manifold is a manifold with additional mathematical properties imposed which permit the definition of compatible coordinate systems on the manifold which are mapped by diffeomorphic functions from the manifold into euclidian n-space.

This mapping is depicted by Figure 1.

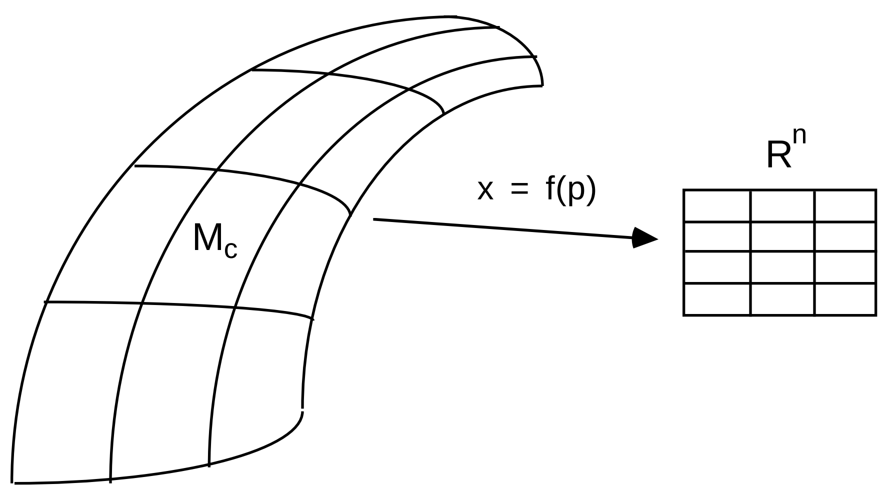

Figure 1
Manifold Patch with Mapping into R^n.

$M_c = \{ p \mid g(p) = c \}$ can be shown to be a C^k differentiable manifold.

Such a manifold may be called a "constraint manifold" since it is totally defined given a set of C^k constraint functions.

The nonlinear programming problem may be stated as

extremalize $f(x)$
over x
subject to $g(x) =$ constant.

For C^k functions f and g with $k > 0$ and for the Jacobian matrix

$$\partial_x g = \begin{bmatrix} \partial_1 g_1 \ldots \partial_n g_1 \\ \ldots \quad \ldots \quad \ldots \\ \partial_1 g_m \ldots \partial_n g_m \end{bmatrix}$$

where $\partial_j g_i = \dfrac{\partial g_i}{\partial x_j}$,

there exist tensors N_p and T_p at the point p defined by

$$N_p = \partial^T{}_p g \, (\partial_p g \, \partial^T{}_p g)^{-1} \, \partial_p g \quad \text{and} \quad T_p = I - N_p$$

where $\partial^T{}_p g$ is the transpose of $\partial_p g$ and the subscript p represents evaluation of the functions at p.

Geometrically, T_p projects onto the tangent space to the constraint manifold at p, and N_p projects onto the normal space to the constraint manifold at p.

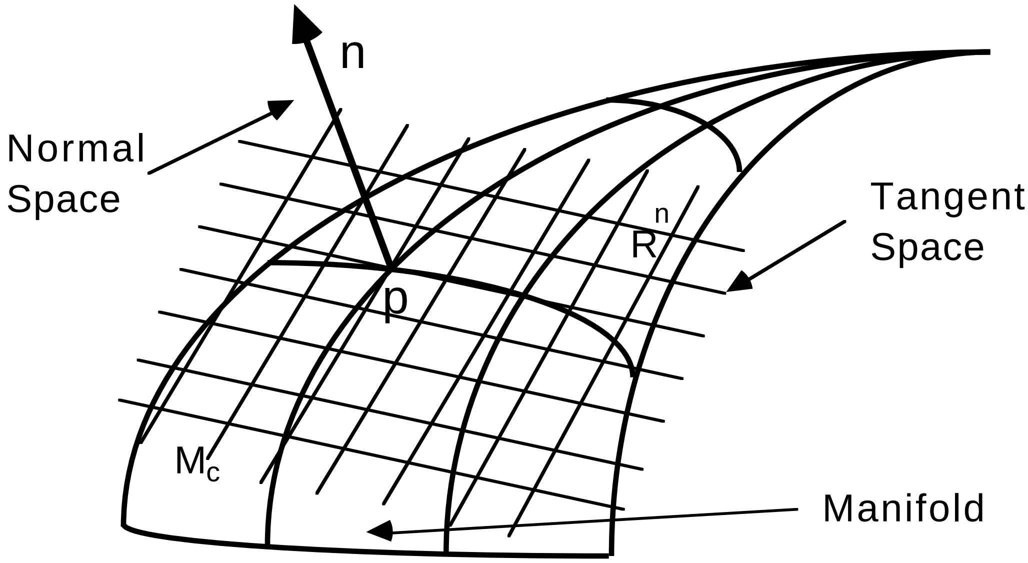

Figure 2
Normal and Tangent Spaces to the Manifold
at the Tensor Operating Point p.

A manifold M_c is covered by a vector field if and only if at each point p there is a vector v(p). If f(x) is C^k function over M_c then the gradient $\partial_p f$ forms a C^k vector field over M_c. If f(x) and g(x) are C^k functions over M_c then $\partial_p f\, T_p$ forms a C^k vector field over M_c.

It is important to note that $\partial_p f\, T_p$ is the restriction of $\partial_p f$ to M_c. That means that $\partial_p f\, T_p$ is in the tangent space to M_c with origin at p.

Tanabe [4] has shown that under appropriate second order conditions, the flow

$$\dot{p} = \pm\, \partial_p f\, T_p$$

extremalizes f(x) on M_c by converging to a local extremum at steady state.

A flow is a geodesic on a manifold M_c if and only if all acceleration is normal to the manifold.

The unit velocity flow

$$\dot{p} = \frac{\pm\partial_p f T_p}{\sqrt{\partial_p f T_p \partial^T{}_p f}}.$$

on M_c is thus a geodesic.

A trivial extension of Tanabe's results [4] shows that under appropriate second order conditions the unit velocity flow extremalizes f(x) on M_c by converging to a local extremum at steady state.

The Eigenvalue/Eigenvector Optimization Problem

For the remainder of this paper, the subscript denoting the tensor operating point will be dropped but it will still be assumed that the point exists and is used.

Let $\Psi_1 \ldots \Psi_n$ be the eigenvectors of a matrix A with eigenvalues λ_{ii}. Then for

$$\Psi = \begin{bmatrix} \psi_1 \\ \ldots \\ \psi_n \end{bmatrix} \text{ and diagonal matrix } \Lambda = \begin{bmatrix} \lambda_{11} \ldots 0 \\ \ldots \ldots \ldots \\ 0 \ \ldots \lambda_{nn} \end{bmatrix},$$

we have

$$\Psi A = \Lambda \Psi \text{ with } \Psi \Psi^T = \Psi^T \Psi = I.$$

Note that

$$A = \Psi^T \Lambda \Psi = \sum_i \lambda_{ii} . \psi_i{}^T \psi_i$$

has components λ_{ii} in the tensor basis matrices $\psi_i{}^T \psi_i$.

The eigenvalues λ_{ii} and eigenvectors ψ_i may be found by solving the following tensor problem.

$$\begin{array}{lllll} \text{Maximum} & \psi_1 A \psi_1{}^T & \psi_2 A \psi_2{}^T & \cdots & \psi_n A \psi_n{}^T \\ \text{subject to} & \psi_1 \psi_1{}^T = 1 & \psi_2 \psi_1{}^T = 0 & \cdots & \psi_n \psi_1{}^T = 0 \end{array}$$

$$\psi_2\ \psi_2^T = 1 \quad \cdots \qquad \psi_n\ \psi_2^T = 0$$
$$\cdots$$
$$\psi_n\ \psi_n^T = 1$$

Suppose that we have $\psi_1, \ldots, \psi_{k-1}$ so far, then we first wish to find a ψ_k which satisfies $\psi_k\ \psi_l^T = 0$, $l = 1, \ldots, k-1$ with $\psi_k\ \psi_k^T = 1$. Then we wish to find the ψ_k such that $\psi_k\ A\ \psi_k^T$ is a maximum over all eligible ψ_k. To find an eligible ψ_k we first choose a random vector y_k and then subtract out components σ_k in the direction of previously found eigenvectors. This ensures orthogonality, that is,

$$(y_k - \sigma_k)\ \left[\psi_1^T \ldots \psi_{k-1}^T\right] = 0.$$

We must now find σ_k. Define

$$H_{k-1} = \left[\psi_1^T \ldots \psi_{k-1}^T\right].$$

Thus

$$y_k\ H_{k-1} = \sigma_k\ H_{k-1}.$$

Let

$$\sigma_k = y_k\ H_{k-1}\ (H_{k-1}^T\ H_{k-1})^{-1}\ H_{k-1}^T\ ,$$

then

$$\sigma_k\ H_{k-1} = y_k\ H_{k-1}\ (H_{k-1}^T\ H_{k-1})^{-1}\ H_{k-1}^T\ H_{k-1} = y\ H_{k-1}$$

as desired. Let

$$Q_{k-1} = H_{k-1}\ (H_{k-1}^T\ H_{k-1})^{-1}\ H_{k-1}^T \text{ and } P_{k-1} = I - Q_{k-1}$$

then

$$\sigma_k = y_k\ Q_{k-1} \qquad \text{and} \qquad y_k - \sigma_k = y_k - y_k\ Q_{k-1} = y_k\ P_{k-1}\ .$$

Let

$$\psi_k = y_k P_{k-1} .$$

Therefore

$$\psi_k H_{k-1} = y_k H_{k-1} - y_k Q_{k-1} H_{k-1} = y_k H_{k-1} - y_k H_{k-1} = 0$$

as desired.

The optimization problem becomes

$$\begin{array}{lllll} \text{Max} & y_1 P_0 A P_0 y_1^T & y_2 P_1 A P_1 y_2^T & \cdots & y_n P_{n-1} A P_{n-1} y_n^T \\ \text{s.t.} & y_1 P_0 y_1^T = 1 & y_2 P_1 y_2^T = 1 & \cdots & y_n P_{n-1} y_n^T = 1 . \end{array}$$

Each subproblem is of the form

$$\begin{array}{ll} \text{Max} & y_k P_{k-1} A P_{k-1} y_k^T \\ \text{s.t.} & y_k P_{k-1} y_k^T = 1 . \end{array}$$

for which

$$\partial_p g = 2 y_1 P_{k-1}$$

and

$$\partial_p g \ \partial_p g^T = 2 y_1 P_{k-1} P_{k-1}^T y_1^T 2 = 4 y_1 P_{k-1} y_1^T = 4 .$$

Thus

$$N_k = \partial^T{}_p g \ (\partial_p g \ \partial^T{}_p g)^{-1} \ \partial_p g$$

$$= \partial^T{}_p g \ (4)^{-1} \ \partial_p g$$

$$= \partial^T{}_p g \ \partial_p g \ /4$$

$$= (2 P_{k-1} y_k^T y_k P_{k-1} 2) /4$$

$$= P_{k-1} y_k^T y_k P_{k-1}$$

and

$$T_k = I - N_k = I - P_{k-1}\, y_k^T\, y_k\, P_{k-1}$$

Thus

$$\dot{y}_k = \partial_y f_k\, T_k$$

$$= 2\, y_k\, P_{k-1}\, A\, P_{k-1}\, (\, I - P_{k-1}\, y_k^T\, y_k\, P_{k-1}\,)$$

$$= 2\, y_k\, P_{k-1}\, A\, P_{k-1} - 2\, y_k\, P_{k-1}\, A\, P_{k-1}\, y_k^T\, y_k\, P_{k-1}$$

Let

$$\lambda_{kk} = y_k\, P_{k-1}\, A\, P_{k-1}\, y_k^T \text{ or}$$

$$\lambda_{kk} = \psi_k\, A\, \psi_k^T,$$

then

$$\dot{y}_k = 2\, y_k\, P_{k-1}\, A\, P_{k-1} - 2\, \lambda_{kk}\, y_k\, P_{k-1}\ .$$

For

$$\dot{y}_k = 0$$

we have

$$y_k\, P_{k-1}\, A\, P_{k-1} = y_k\, P_{k-1}\, \lambda_{kk} \text{ or}$$

$$\psi_k\, A\, P_{k-1} = \psi_k\, \lambda_{kk}$$

where λ_{kk} and ψ_k are the eigenvalue and eigenvector of $A\, P_{k-1}$. The question becomes, is

$$\psi_k\, A = \psi_k\, \lambda_{kk}$$

also. Suppose that is not true, then

$$\psi_k\, A \neq \psi_k\, \lambda_{kk}$$

implies that

$$\psi_k A P_{k-1} \neq \psi_k P_{k-1} \lambda_{kk} = y_k P_{k-1} P_{k-1} \lambda_{kk} = y_k P_{k-1} \lambda_{kk} = \psi_k \lambda_{kk}$$

which is false. Thus

$$\psi_k A = \psi_k \lambda_{kk} .$$

The Algorithm Structure

To start the problem we set $P_0 = I$ and choose y_1 such that $y_1 P_0 y_1^T = 1$, form P_1 from $\psi_1 = y_1 P_0$, choose y_2 such that $y_2 P_1 y_2^T = 1$ and calculate $\psi_2 = y_2 P_1$, form P_2 from ψ_1, ψ_2 and so on until we get $\psi_n = y_n P_n$. With the problem properly initialized we begin the simultaneous solution of the n subproblems

$$\lambda_{kk} = y_k P_{k-1} A P_{k-1} y_k^T$$

$$\dot{y}_k = 2 y_k P_{k-1} A P_{k-1} - 2 \lambda_{kk} y_k P_{k-1}$$

for which λ_{kk} is the current approximation of the kth eigenvalue and

$$\psi_k = y_k P_{k-1}$$

is the current approximation of the kth eigenvector.

The Dynamic Case

The dynamic capabilities to track the eigenvalues and eigenvectors become obvious from the algorithm structure. If $A = A(t)$ and if

$$\dot{y}_k(t) = 0,$$

then at $t + \delta t$ if

$$A(t + \delta t) \neq A(t) \text{ then}$$

$$\dot{y}_k(t + \delta t) \neq 0.$$

This causes the dynamic process to converge toward the new eigenvalues and eigenvectors. It does not, however, guarantee that the the new eigenvalues and eigenvectors will be attained. The practical case sees a tracking lag resulting in a perpetual error until A(t) becomes constant.

Conclusions

From a computational viewpoint, the simultaneous solution of the resulting n^2 components can present a problem. However, this can be alleviated memory wise and speed wise somewhat by solving first for the largest eigenvalue, then solving for the next largest and so on.

It is most interesting to note, however, that not only can continuous optimization on manifolds provide dynamic numerical and closed form solutions to optimization problems [1], but can also be used to derive closed form dynamic algorithm structures. The author has on a number of other occasions applied this cabability of COCM to derive dynamical systems corresponding to specific problems and then use these systems to obtain a better theoretical understanding of the problem.

References

[1] Dean, E., "Continuous Optimization on Constraint Manifolds," presented at the TIMS/ORSA Joint National Meeting, April 25-27, 1988.

[2] Tanabe, K., "Algorithm for the Constrained Maximization Technique for Nonlinear Programming," *Proceedings of the Second Hawaii International Conference on Systems Science*, 1969, 487-490.

[3] Luenberger, D., "The Gradient Projection Method along Geodesics," *Management Science*, Vol 18, No. 11, July 1972.

[4] Tanabe, K., "An Algorithm for the Constrained Maximization in Nonlinear Programming," *J. Operations Research Soc. of Japan*, Vol. 17, No. 4, December 1974.

[5] Tanabe, K., "A Geometric Method in Nonlinear Programming," Stanford University Computer Science Department, Report No. STAN-CS-643, 1977.

[6] Tanabe, K., "A Geometric Method in Nonlinear Programming," *Journal of Optimization Theory and Applications*, Vol 30, No. 2, February 1980, pp181-210.

[7] Rapcsák, T., "Minimum Problems on Differentiable Manifolds," working paper, Computer and Automation Institute, Hungarian Academy of Sciences, Hungary, Budapest, P.O. Box 63 H-1502, November, 1986.

[8] Gerretsen, J. C. H., Lectures on Tensor Calculus and Differential Geometry, P. Noordhoff, Groningen, The Netherlands, 1962.

[9] Thorpe, J. A., Elementary Topics in Differential Geometry, Springer-Verlag, New York, NY, 1979.

[10] Boothby, W. M., An Introduction to Differentiable Manifolds and Riemannian Geometry, Academic Press, New York, NY, 1975.